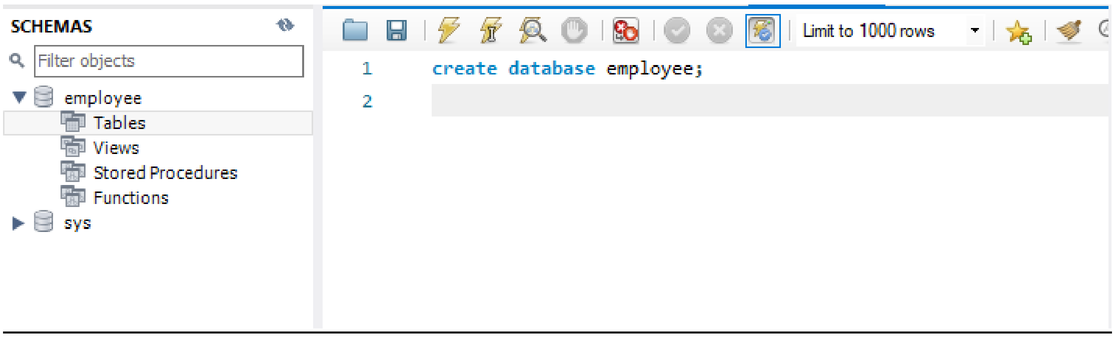
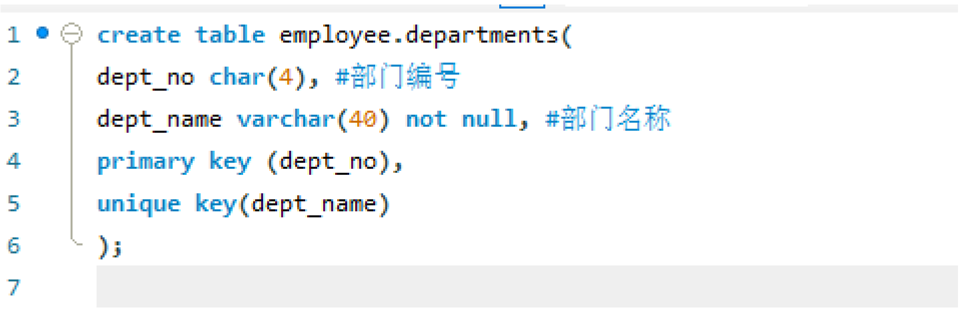
实验一结果截图

一、实验截图

（注意截图清晰，包含完整的 sql 语句和运行结果）



```sql
create database employee;
```



```sql
create table employee.departments(
dept_no char(4), #部门编号
dept_name varchar(40) not null, #部门名称
primary key (dept_no),
unique key(dept_name)
);
```

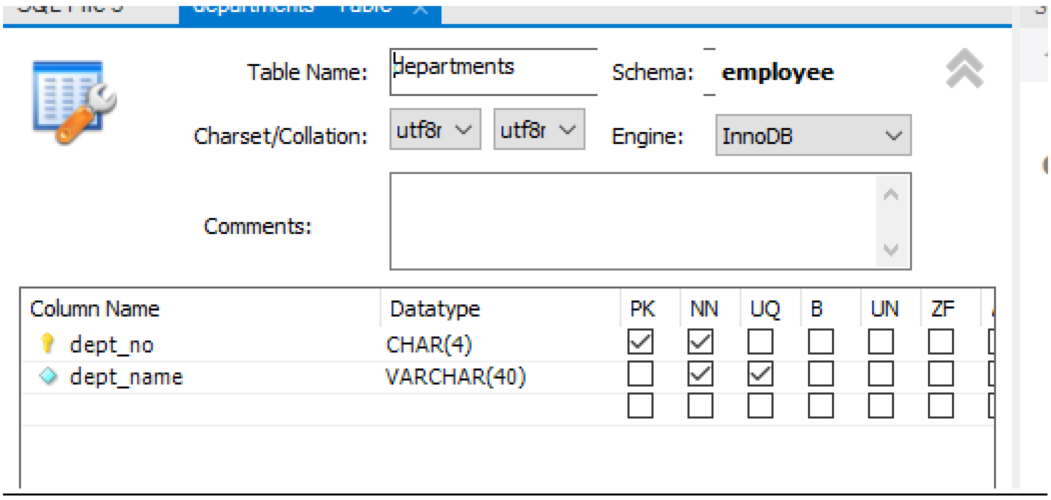
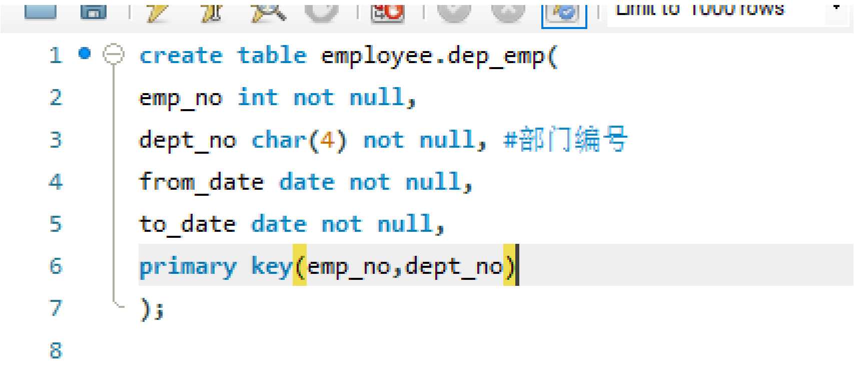


Table Name: departments
Schema: employee
Charset/Collation: utf8r utf8r
Engine: InnoDB
Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF
dept_no	CHAR(4)	☑	☑	□	□	□	□
dept_name	VARCHAR(40)	□	☑	☑	□	□	□
		□	□	□	□	□	□



```
create table employee.dep_emp(
emp_no int not null,
dept_no char(4) not null, #部门编号
from_date date not null,
to_date date not null,
primary key(emp_no,dept_no)
);
```

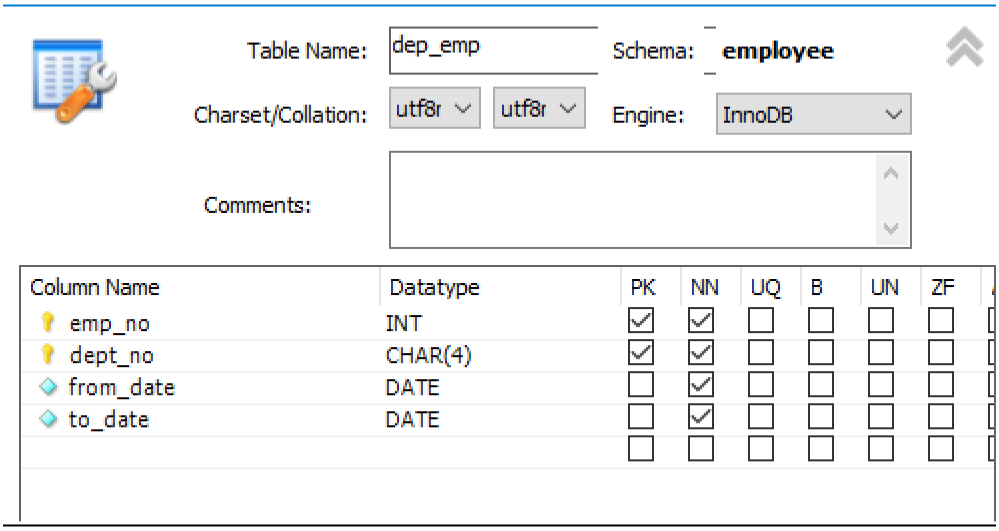


Table Name: dep_emp

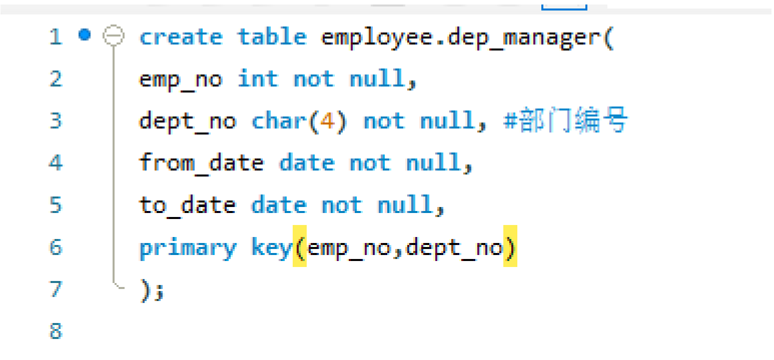
Schema: employee

Charset/Collation: utf8r utf8r

Engine: InnoDB

Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF
emp_no	INT	☑	☑	□	□	□	□
dept_no	CHAR(4)	☑	☑	□	□	□	□
from_date	DATE	□	☑	□	□	□	□
to_date	DATE	□	☑	□	□	□	□
		□	□	□	□	□	□



```
create table employee.dep_manager(
emp_no int not null,
dept_no char(4) not null, #部门编号
from_date date not null,
to_date date not null,
primary key(emp_no,dept_no)
);
```

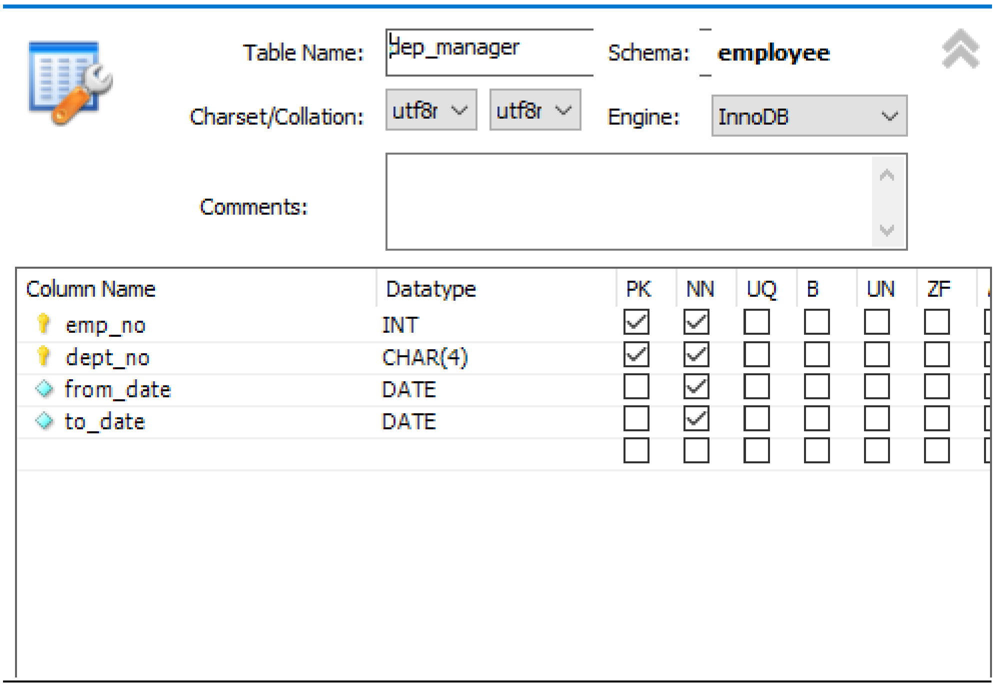


Table Name: dep_manager

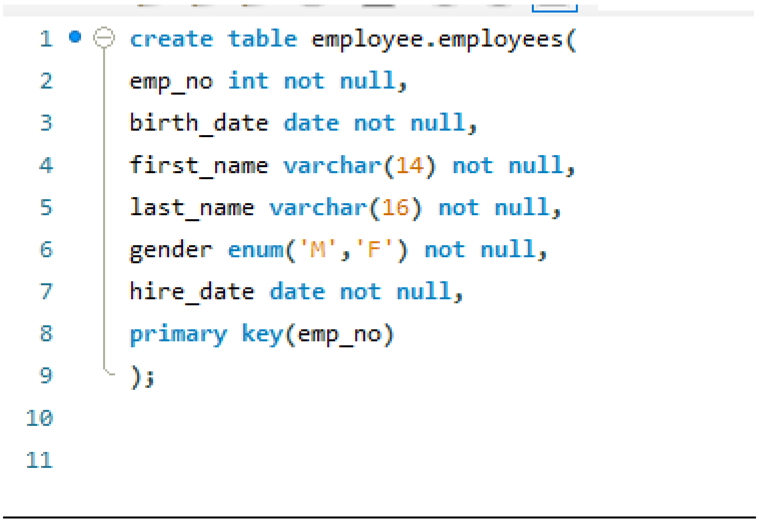
Schema: employee

Charset/Collation: utf8r utf8r

Engine: InnoDB

Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF
emp_no	INT	☑	☑	□	□	□	□
dept_no	CHAR(4)	☑	☑	□	□	□	□
from_date	DATE	□	☑	□	□	□	□
to_date	DATE	□	☑	□	□	□	□
		□	□	□	□	□	□



```
create table employee.employees(
emp_no int not null,
birth_date date not null,
first_name varchar(14) not null,
last_name varchar(16) not null,
gender enum('M','F') not null,
hire_date date not null,
primary key(emp_no)
);


```

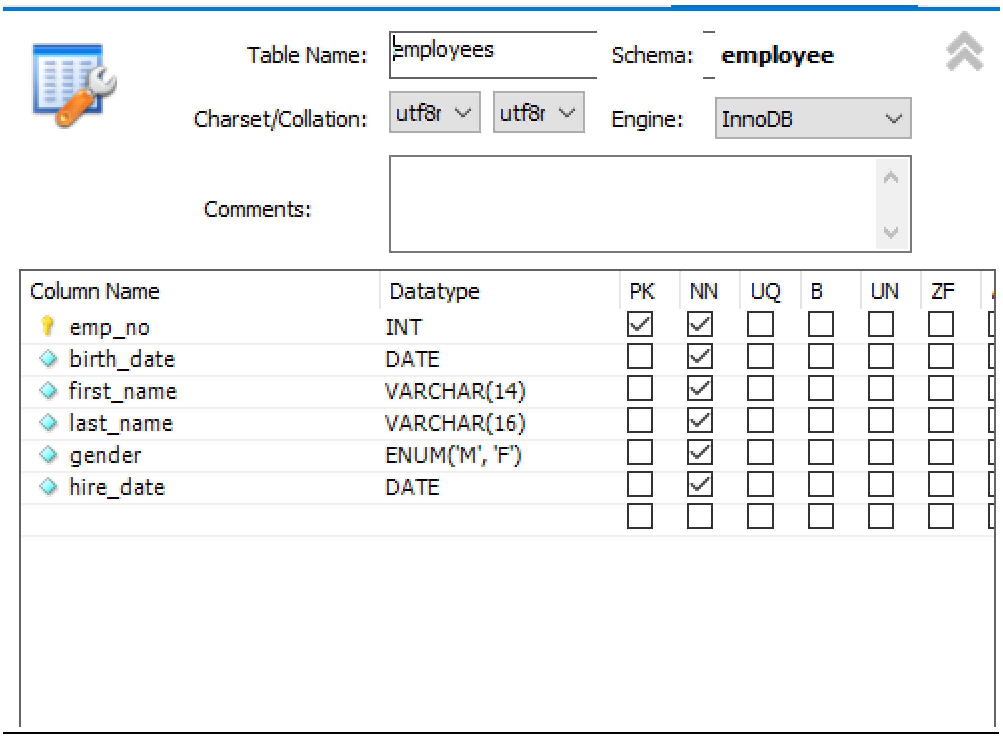
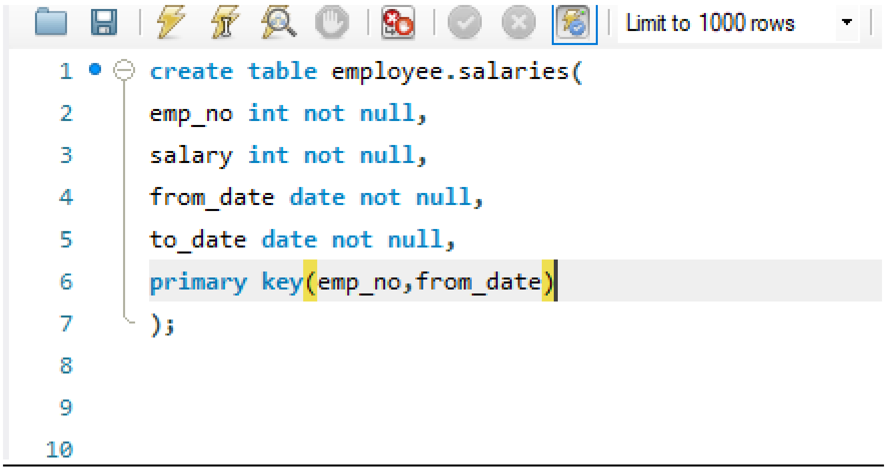


Table Name: employees
Schema: employee
Charset/Collation: utf8r utf8r
Engine: InnoDB
Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF
emp_no	INT	☑	☑	□	□	□	□
birth_date	DATE	□	☑	□	□	□	□
first_name	VARCHAR(14)	□	☑	□	□	□	□
last_name	VARCHAR(16)	□	☑	□	□	□	□
gender	ENUM('M', 'F')	□	☑	□	□	□	□
hire_date	DATE	□	☑	□	□	□	□
		□	□	□	□	□	□



Limit to 1000 rows

```
create table employee.salaries(
emp_no int not null,
salary int not null,
from_date date not null,
to_date date not null,
primary key(emp_no,from_date)
);
```

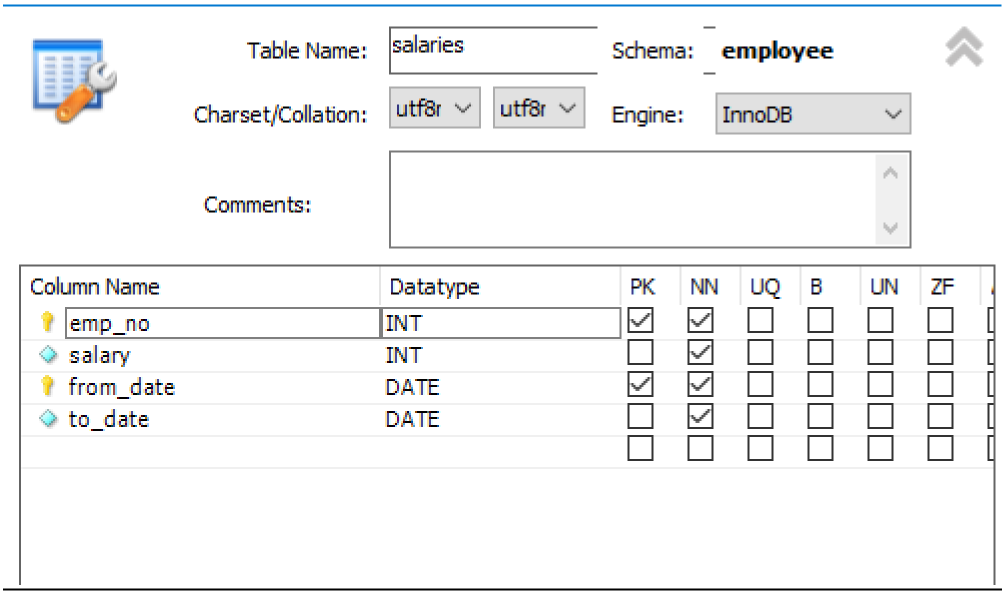
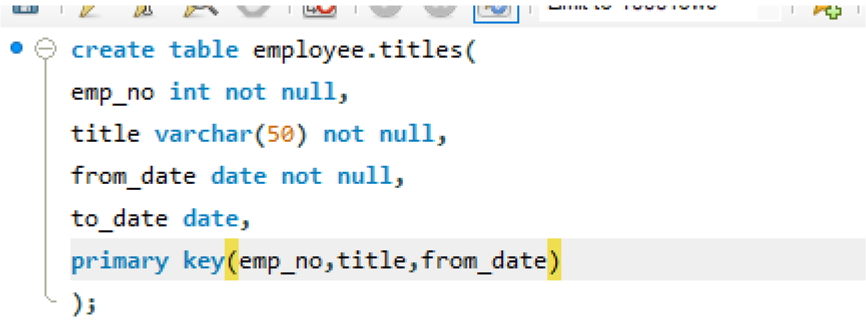


Table Name: salaries
Schema: employee
Charset/Collation: utf8r utf8r
Engine: InnoDB
Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF
emp_no	INT	☑	☑	□	□	□	□
salary	INT	□	☑	□	□	□	□
from_date	DATE	☑	☑	□	□	□	□
to_date	DATE	□	☑	□	□	□	□
		□	□	□	□	□	□



```
create table employee.titles(
emp_no int not null,
title varchar(50) not null,
from_date date not null,
to_date date,
primary key(emp_no,title,from_date)
);
```

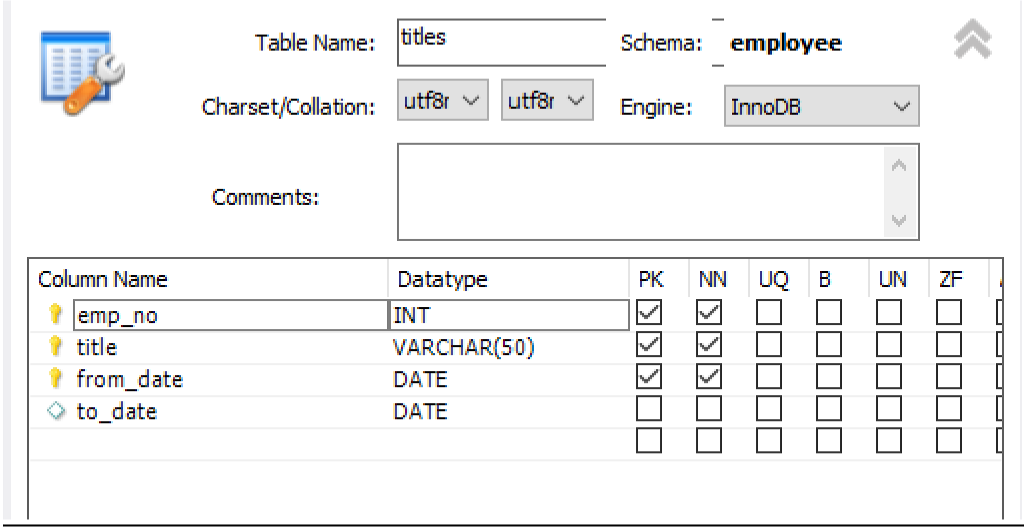


Table Name: titles

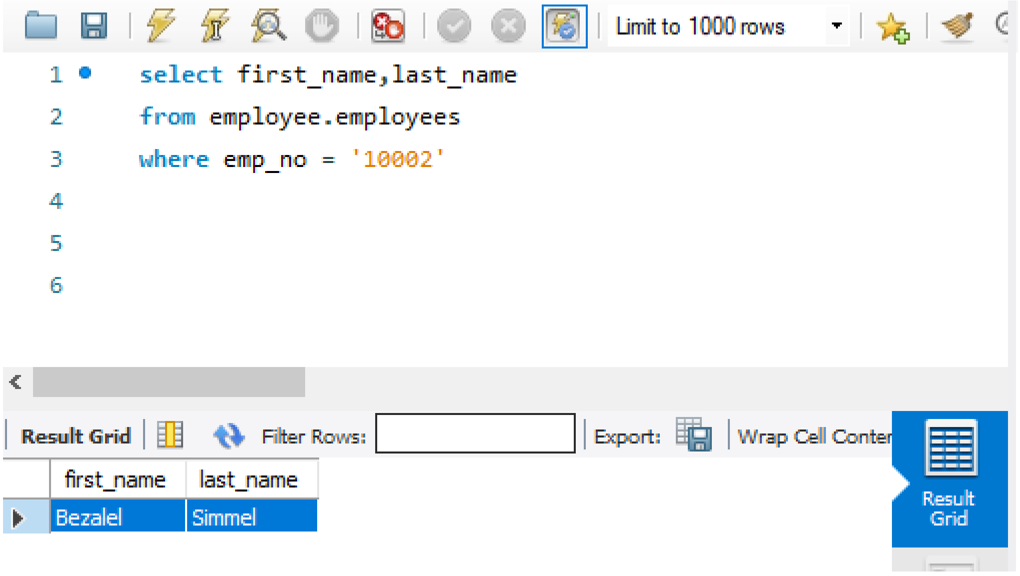
Schema: employee

Charset/Collation: utf8r utf8r

Engine: InnoDB

Comments:

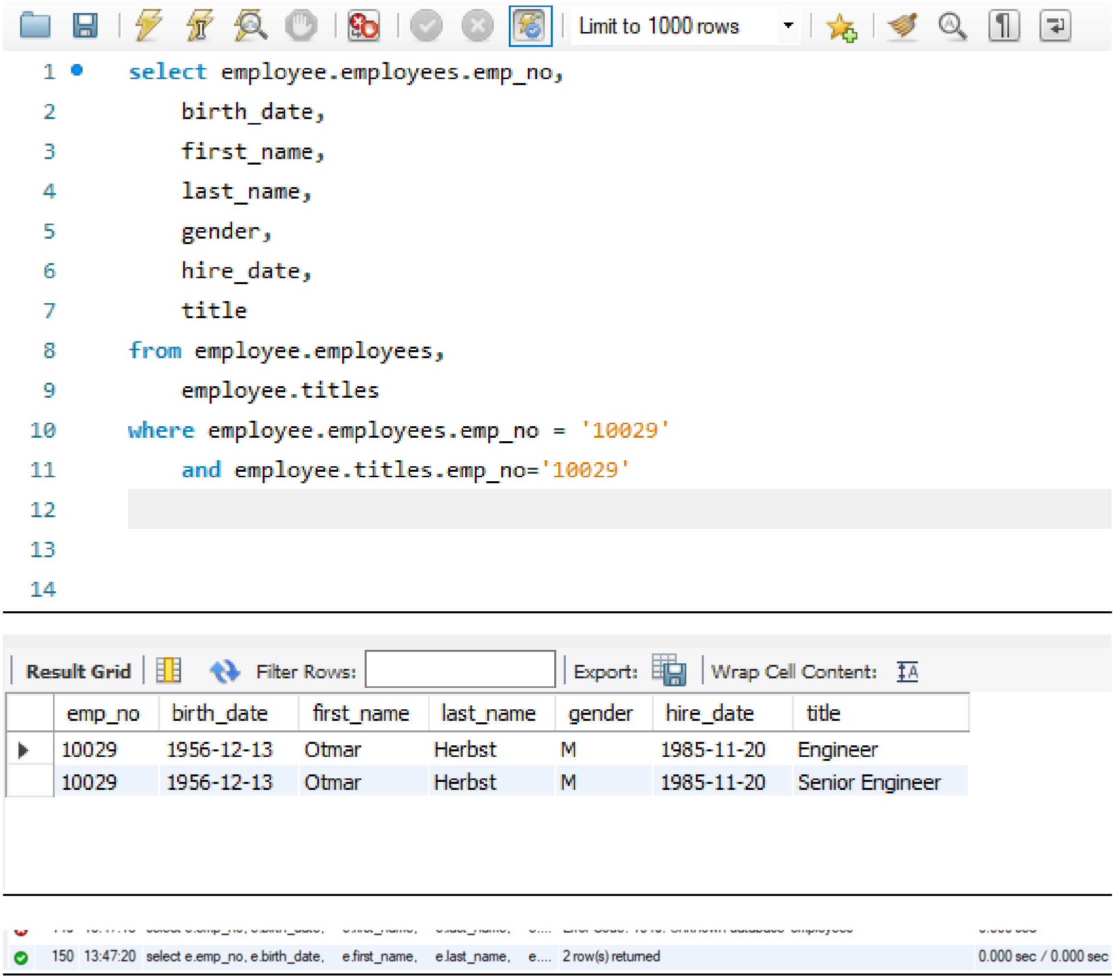
Column Name	Datatype	PK	NN	UQ	B	UN	ZF
emp_no	INT	☑	☑	□	□	□	□
title	VARCHAR(50)	☑	☑	□	□	□	□
from_date	DATE	☑	☑	□	□	□	□
to_date	DATE	□	□	□	□	□	□
		□	□	□	□	□	□



```
select first_name,last_name
from employee.employees
where emp_no = '10002'
```

first_name	last_name
Bezalel	Simmel

1. 查询 emp_no 是“10029”的员工信息，显示其 emp_no, birth_date, first_name, last_name, gender, hire_date, title；

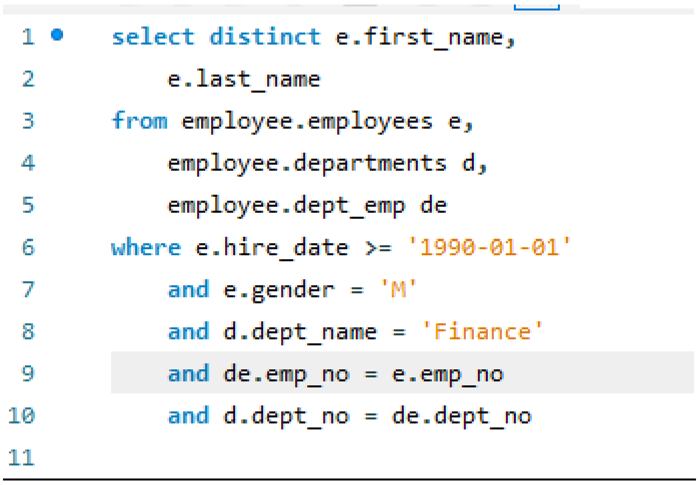


```sql
select employee.employees.emp_no,
    birth_date,
    first_name,
    last_name,
    gender,
    hire_date,
    title
from employee.employees,
    employee.titles
where employee.employees.emp_no = '10029'
    and employee.titles.emp_no='10029'
```

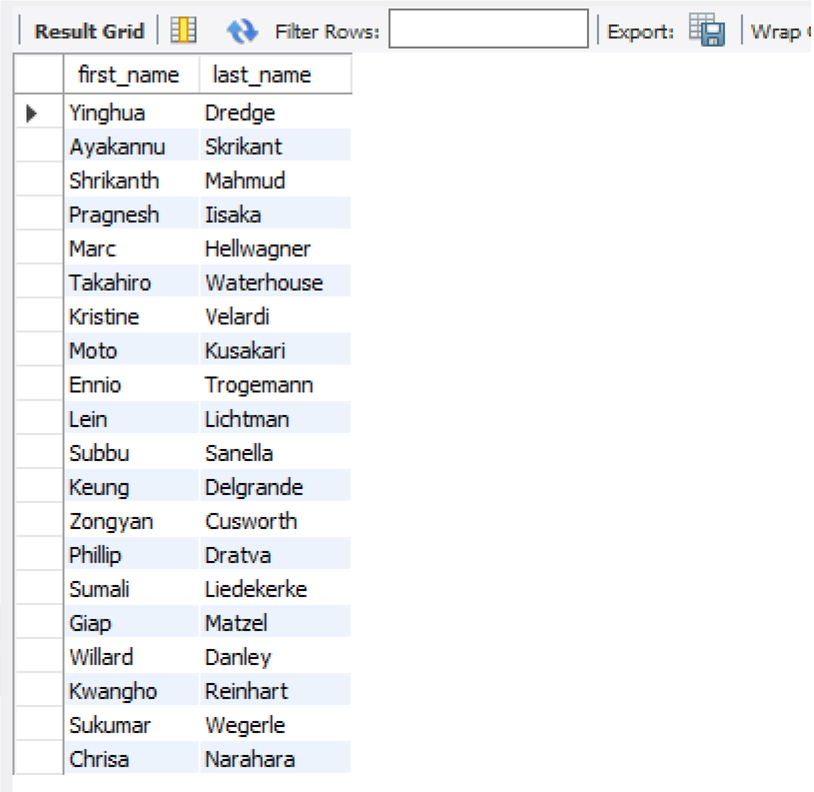
emp_no	birth_date	first_name	last_name	gender	hire_date	title
10029	1956-12-13	Otmar	Herbst	M	1985-11-20	Engineer
10029	1956-12-13	Otmar	Herbst	M	1985-11-20	Senior Engineer

```
150  13:47:20  select e.emp_no, e.birth_date,   e.first_name,   e.last_name,   e....  2 row(s) returned        0.000 sec / 0.000 sec
```

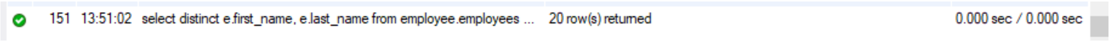
2. 查询入职时间在 1990 年后且在“Finance”部门工作过的男员工姓名；



```sql
select distinct e.first_name,
    e.last_name
from employee.employees e,
    employee.departments d,
    employee.dept_emp de
where e.hire_date >= '1990-01-01'
    and e.gender = 'M'
    and d.dept_name = 'Finance'
    and de.emp_no = e.emp_no
    and d.dept_no = de.dept_no
```

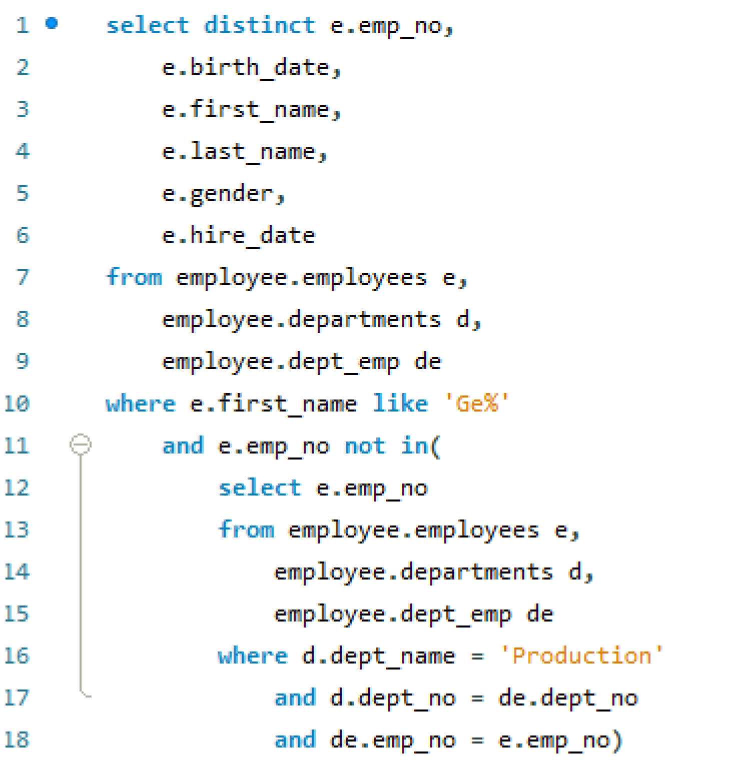


first_name	last_name
Yinghua	Dredge
Ayakannu	Skrikant
Shrikanth	Mahmud
Pragnesh	Iisaka
Marc	Hellwagner
Takahiro	Waterhouse
Kristine	Velardi
Moto	Kusakari
Ennio	Trogemann
Lein	Lichtman
Subbu	Sanella
Keung	Delgrande
Zongyan	Cusworth
Phillip	Dratva
Sumali	Liedekerke
Giap	Matzel
Willard	Danley
Kwangho	Reinhart
Sukumar	Wegerle
Chrisa	Narahara

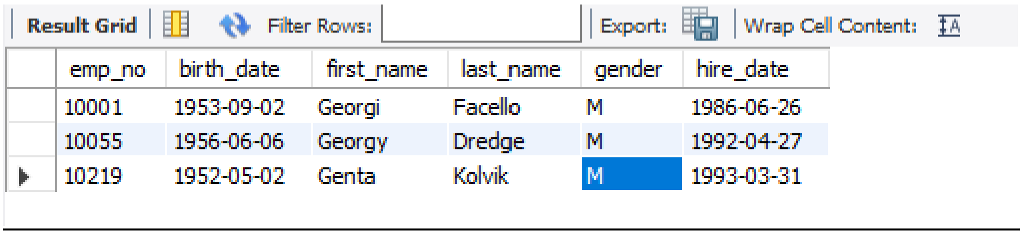


```
151  13:51:02  select distinct e.first_name, e.last_name from employee.employees ...   20 row(s) returned   0.000 sec / 0.000 sec
```

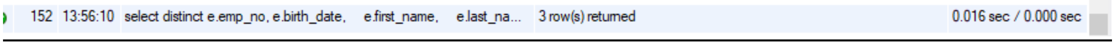
3. 查询没有在“Production”部门工作过且 first_name 是“Ge”开头的的员工信息，显示其 emp_no, birth_date, first_name, last_name, gender, hire_date；



```sql
select distinct e.emp_no,
    e.birth_date,
    e.first_name,
    e.last_name,
    e.gender,
    e.hire_date
from employee.employees e,
    employee.departments d,
    employee.dept_emp de
where e.first_name like 'Ge%'
    and e.emp_no not in(
        select e.emp_no
        from employee.employees e,
            employee.departments d,
            employee.dept_emp de
        where d.dept_name = 'Production'
            and d.dept_no = de.dept_no
            and de.emp_no = e.emp_no)
```

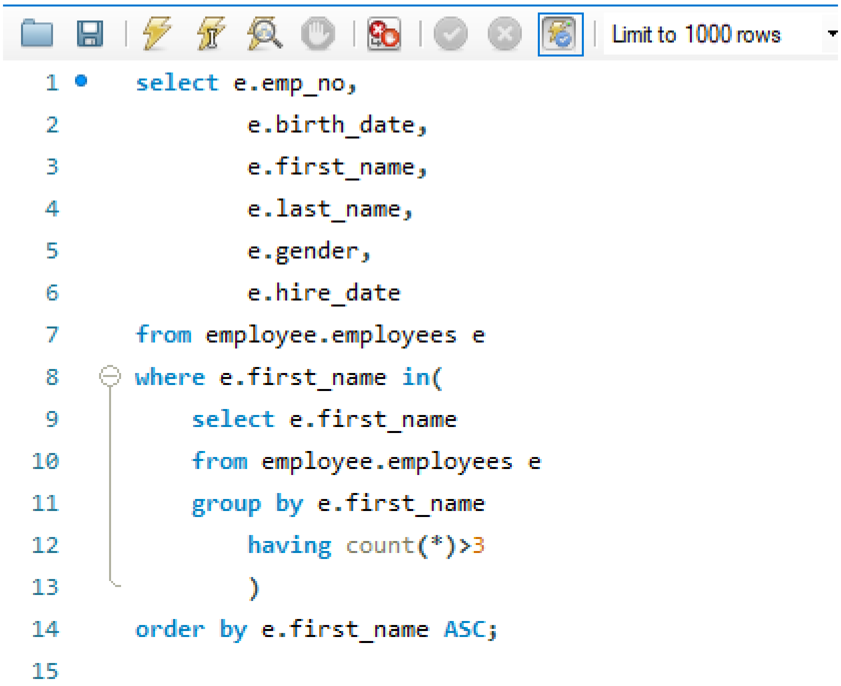


emp_no	birth_date	first_name	last_name	gender	hire_date
10001	1953-09-02	Georgi	Facello	M	1986-06-26
10055	1956-06-06	Georgy	Dredge	M	1992-04-27
10219	1952-05-02	Genta	Kolvik	M	1993-03-31

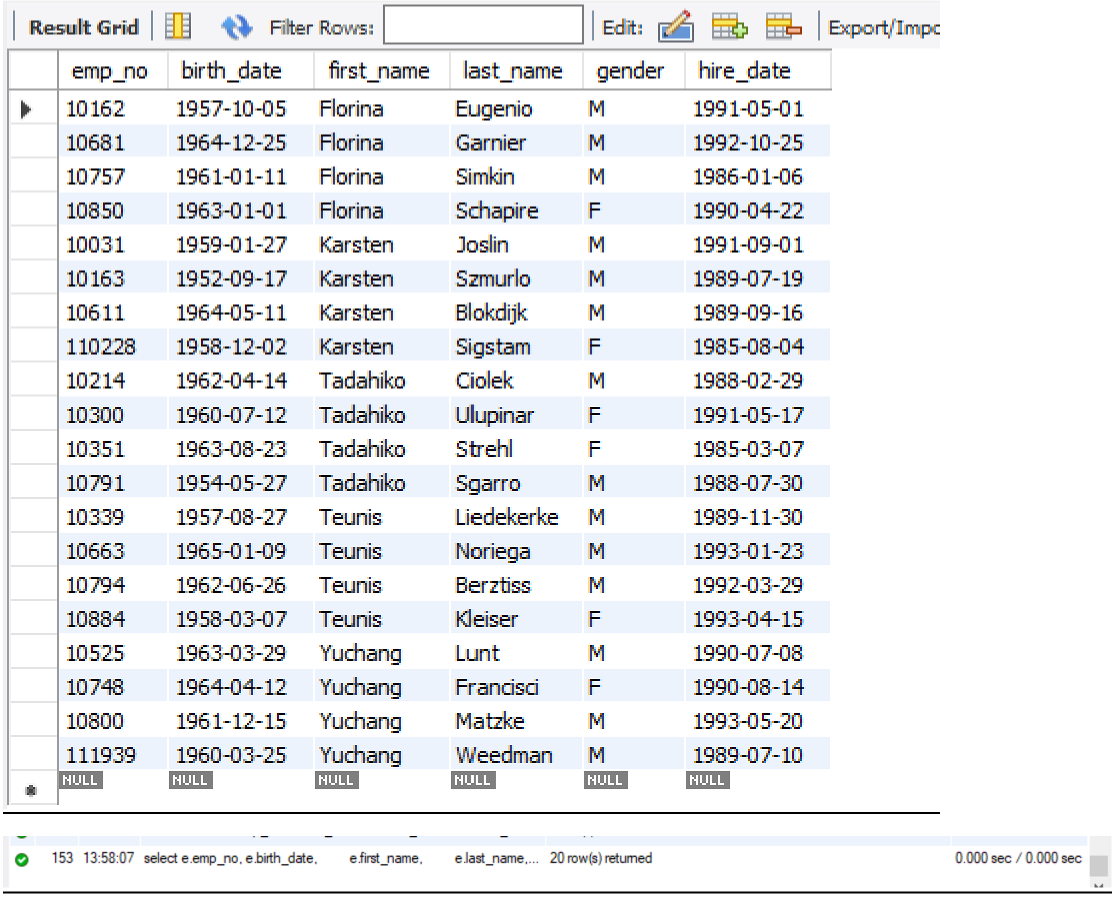


```
152  13:56:10  select distinct e.emp_no, e.birth_date,    e.first_name,    e.last_na...  3 row(s) returned    0.016 sec / 0.000 sec
```

4. 查询 first_name 相同且人数超过 3 人的员工信息，显示其 emp_no, birth_date, first_name, last_name, gender, hire_date，要求按 first_name 升序显示；



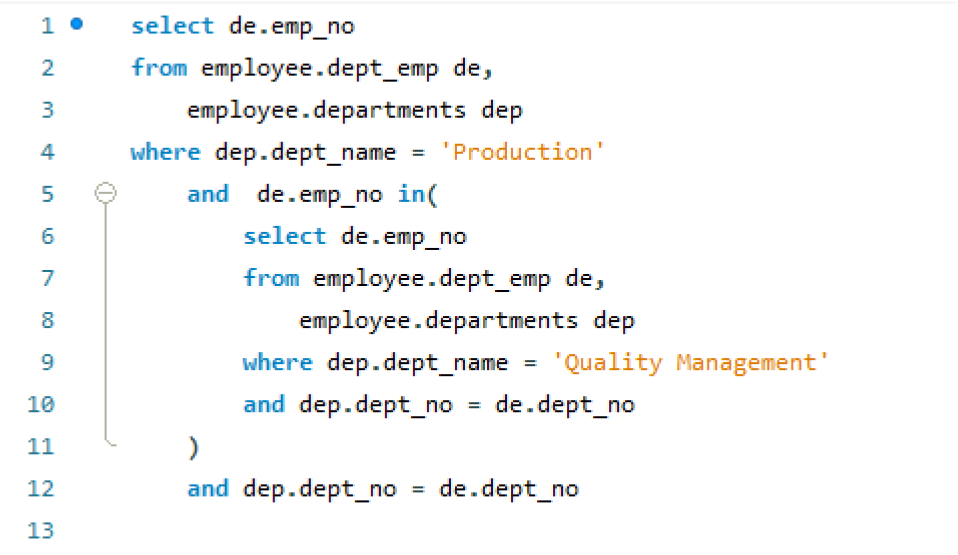
```sql
select e.emp_no,
        e.birth_date,
        e.first_name,
        e.last_name,
        e.gender,
        e.hire_date
from employee.employees e
where e.first_name in(
    select e.first_name
    from employee.employees e
    group by e.first_name
        having count(*)>3
        )
order by e.first_name ASC;
```



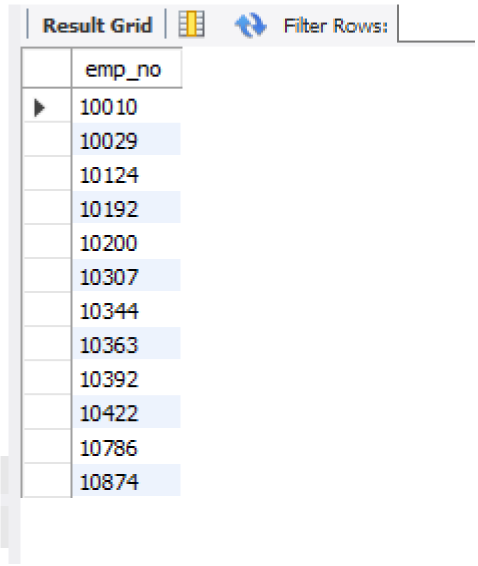
emp_no	birth_date	first_name	last_name	gender	hire_date
10162	1957-10-05	Florina	Eugenio	M	1991-05-01
10681	1964-12-25	Florina	Garnier	M	1992-10-25
10757	1961-01-11	Florina	Simkin	M	1986-01-06
10850	1963-01-01	Florina	Schapire	F	1990-04-22
10031	1959-01-27	Karsten	Joslin	M	1991-09-01
10163	1952-09-17	Karsten	Szmurlo	M	1989-07-19
10611	1964-05-11	Karsten	Blokdijk	M	1989-09-16
110228	1958-12-02	Karsten	Sigstam	F	1985-08-04
10214	1962-04-14	Tadahiko	Ciolek	M	1988-02-29
10300	1960-07-12	Tadahiko	Ulupinar	F	1991-05-17
10351	1963-08-23	Tadahiko	Strehl	F	1985-03-07
10791	1954-05-27	Tadahiko	Sgarro	M	1988-07-30
10339	1957-08-27	Teunis	Liedekerke	M	1989-11-30
10663	1965-01-09	Teunis	Noriega	M	1993-01-23
10794	1962-06-26	Teunis	Berztiss	M	1992-03-29
10884	1958-03-07	Teunis	Kleiser	F	1993-04-15
10525	1963-03-29	Yuchang	Lunt	M	1990-07-08
10748	1964-04-12	Yuchang	Francisci	F	1990-08-14
10800	1961-12-15	Yuchang	Matzke	M	1993-05-20
111939	1960-03-25	Yuchang	Weedman	M	1989-07-10
NULL	NULL	NULL	NULL	NULL	NULL

```
153  13:58:07  select e.emp_no, e.birth_date,    e.first_name,    e.last_name,...  20 row(s) returned        0.000 sec / 0.000 sec
```

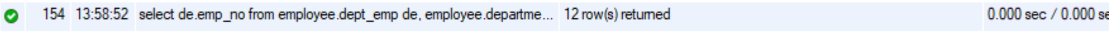
5. 查询至少在“Production”和“Quality Management”两个部门都工作过的员工编号；



```sql
select de.emp_no
from employee.dept_emp de,
    employee.departments dep
where dep.dept_name = 'Production'
    and  de.emp_no in(
        select de.emp_no
        from employee.dept_emp de,
            employee.departments dep
        where dep.dept_name = 'Quality Management'
        and dep.dept_no = de.dept_no
    )
    and dep.dept_no = de.dept_no
```

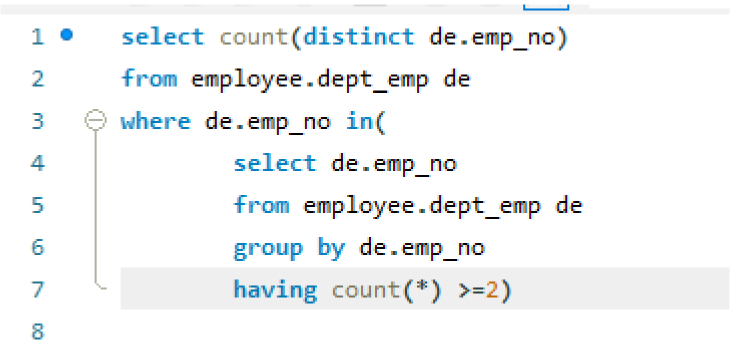


emp_no
10010
10029
10124
10192
10200
10307
10344
10363
10392
10422
10786
10874

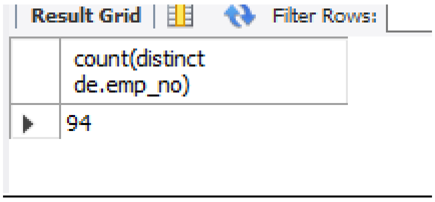


```
154  13:58:52  select de.emp_no from employee.dept_emp de, employee.departme...  12 row(s) returned  0.000 sec / 0.000 se
```

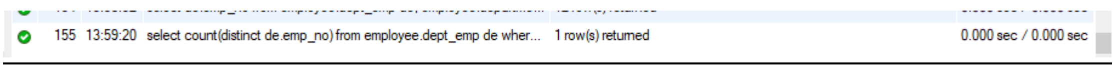
6. 查询至少在 2 个部门工作过的员工人数；



```sql
select count(distinct de.emp_no)
from employee.dept_emp de
where de.emp_no in(
        select de.emp_no
        from employee.dept_emp de
        group by de.emp_no
        having count(*) >=2)
```

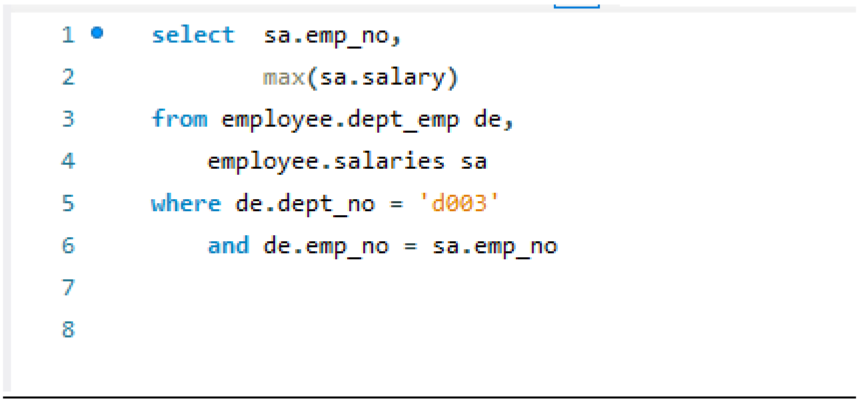


	count(distinct de.emp_no)
▶	94

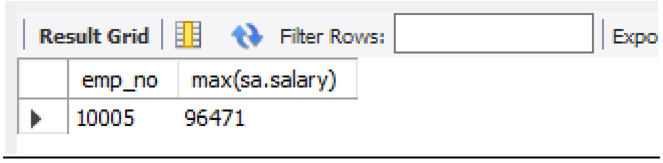


```
155  13:59:20  select count(distinct de.emp_no) from employee.dept_emp de wher...  1 row(s) returned  0.000 sec / 0.000 sec
```

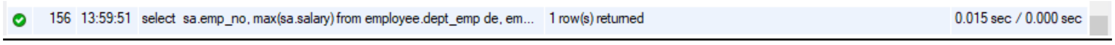
7. 查询在“d003”部门工作过的且工资最高的员工编号及其最高工资；



```
select  sa.emp_no,
        max(sa.salary)
from employee.dept_emp de,
    employee.salaries sa
where de.dept_no = 'd003'
    and de.emp_no = sa.emp_no
```

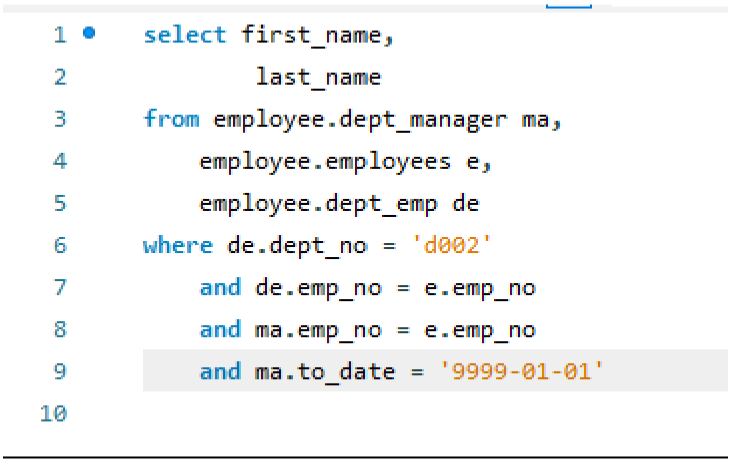


	emp_no	max(sa.salary)
▶	10005	96471

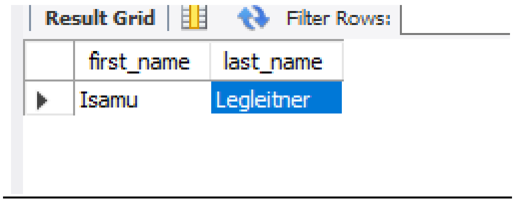


```
156  13:59:51  select  sa.emp_no, max(sa.salary) from employee.dept_emp de, em...  1 row(s) returned  0.015 sec / 0.000 sec
```

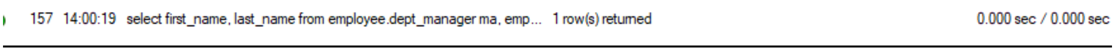
8. 查询“d002”部门的当前领导姓名；



```
select first_name,
        last_name
from employee.dept_manager ma,
    employee.employees e,
    employee.dept_emp de
where de.dept_no = 'd002'
    and de.emp_no = e.emp_no
    and ma.emp_no = e.emp_no
    and ma.to_date = '9999-01-01'
```

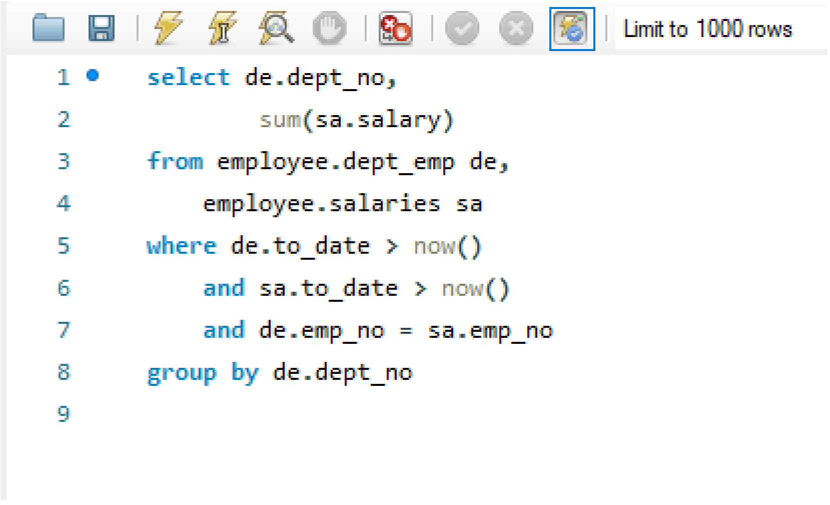


	first_name	last_name
▶	Isamu	Legleitner



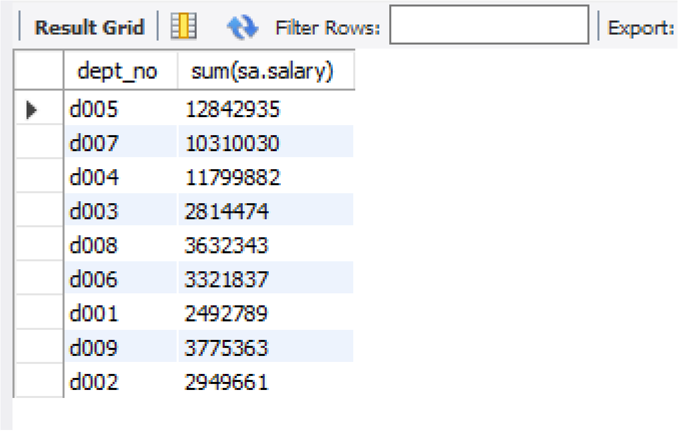
```
157  14:00:19  select first_name, last_name from employee.dept_manager ma, emp...  1 row(s) returned  0.000 sec / 0.000 sec
```

9. 查询当前每个部门的部门编号和员工总工资；

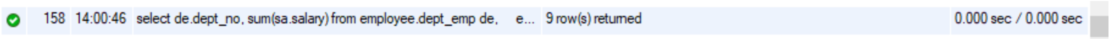


```sql
select de.dept_no,
        sum(sa.salary)
from employee.dept_emp de,
    employee.salaries sa
where de.to_date > now()
    and sa.to_date > now()
    and de.emp_no = sa.emp_no
group by de.dept_no

```

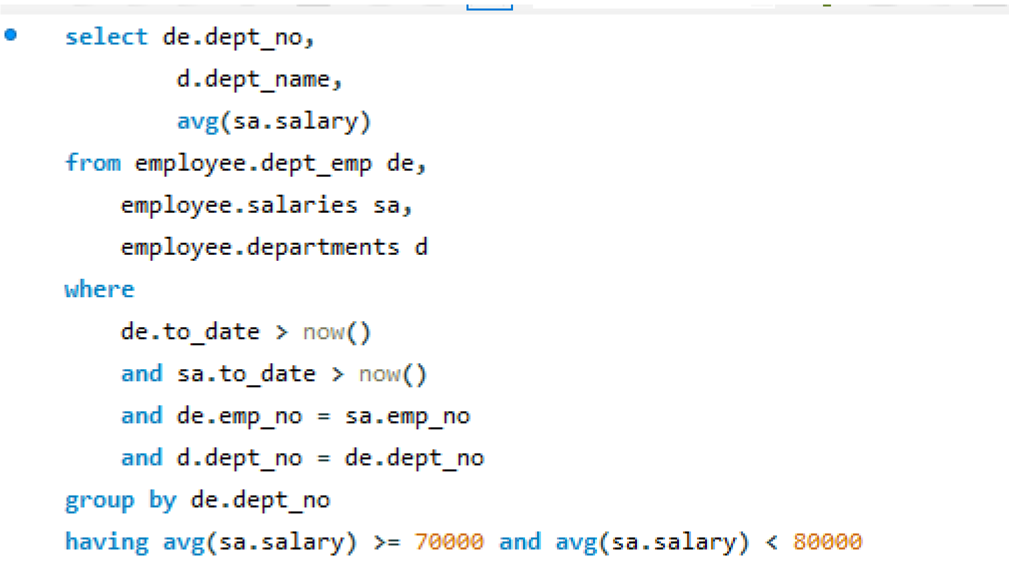


dept_no	sum(sa.salary)
d005	12842935
d007	10310030
d004	11799882
d003	2814474
d008	3632343
d006	3321837
d001	2492789
d009	3775363
d002	2949661

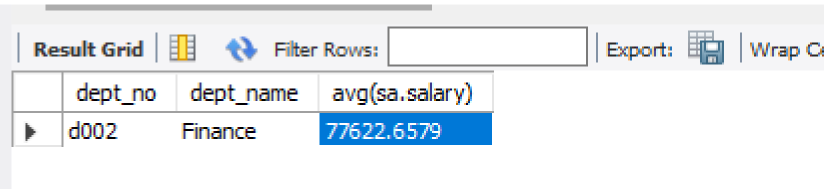


```
158 14:00:46 select de.dept_no, sum(sa.salary) from employee.dept_emp de, e... 9 row(s) returned 0.000 sec / 0.000 sec
```

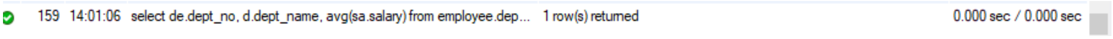
10. 查询当前部门员工平均工资在 70000 元到 80000 元（包含 70000，低于 80000）的部门编号，部门名称和员工平均工资；



```sql
select de.dept_no,
        d.dept_name,
        avg(sa.salary)
from employee.dept_emp de,
    employee.salaries sa,
    employee.departments d
where
    de.to_date > now()
    and sa.to_date > now()
    and de.emp_no = sa.emp_no
    and d.dept_no = de.dept_no
group by de.dept_no
having avg(sa.salary) >= 70000 and avg(sa.salary) < 80000
```

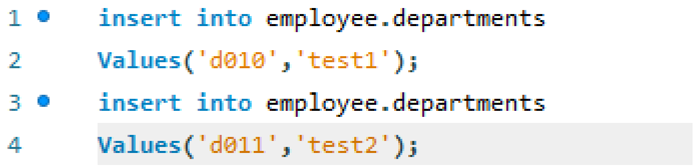


	dept_no	dept_name	avg(sa.salary)
▶	d002	Finance	77622.6579

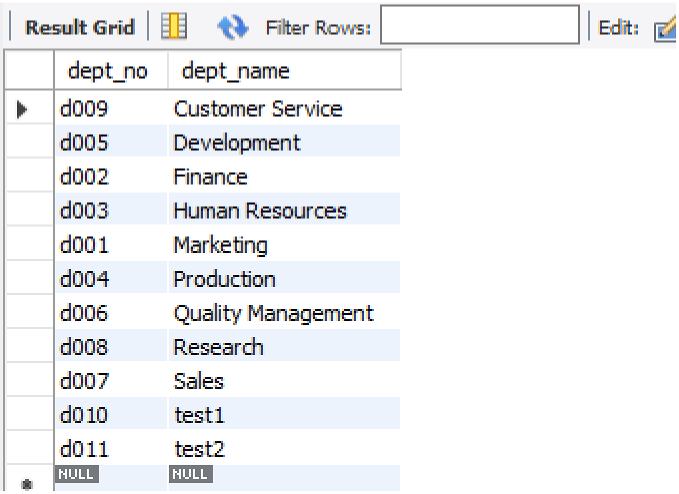


```
159  14:01:06  select de.dept_no, d.dept_name, avg(sa.salary) from employee.dep...  1 row(s) returned    0.000 sec / 0.000 sec
```

11. 在 departments 表新增 2 条记录（内容自定）;

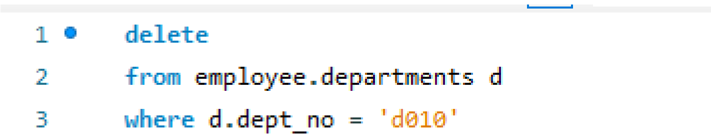


```
insert into employee.departments
Values('d010','test1');
insert into employee.departments
Values('d011','test2');
```

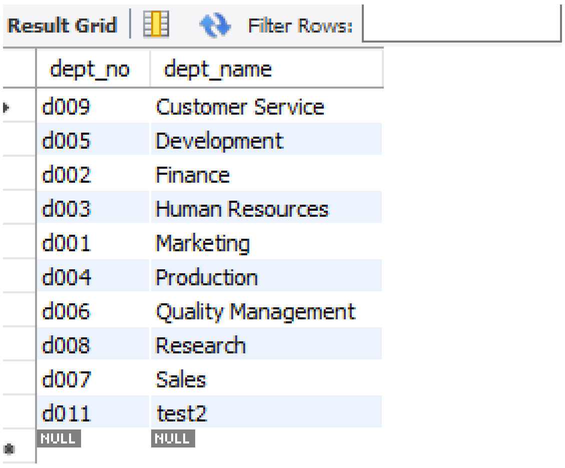


	dept_no	dept_name
▶	d009	Customer Service
	d005	Development
	d002	Finance
	d003	Human Resources
	d001	Marketing
	d004	Production
	d006	Quality Management
	d008	Research
	d007	Sales
	d010	test1
	d011	test2
●	NULL	NULL

12. 在 departments 表中删除刚才新增的 2 条记录中的 1 条;

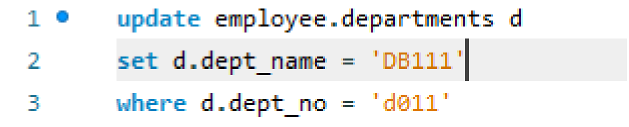


```
delete
from employee.departments d
where d.dept_no = 'd010'
```

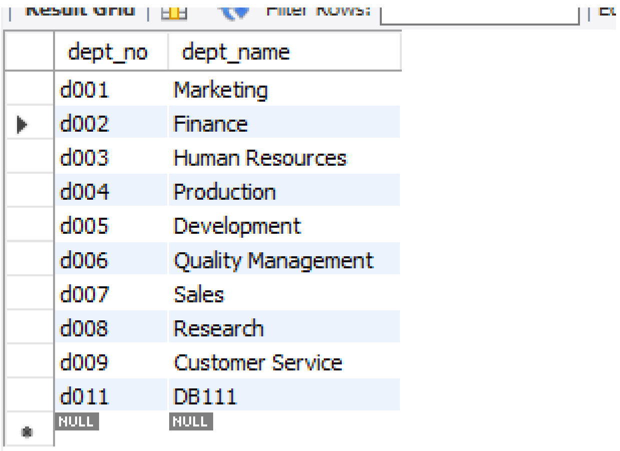


	dept_no	dept_name
▶	d009	Customer Service
	d005	Development
	d002	Finance
	d003	Human Resources
	d001	Marketing
	d004	Production
	d006	Quality Management
	d008	Research
	d007	Sales
	d011	test2
●	NULL	NULL

13. 在 departments 表中修改步骤 11 新增的记录；

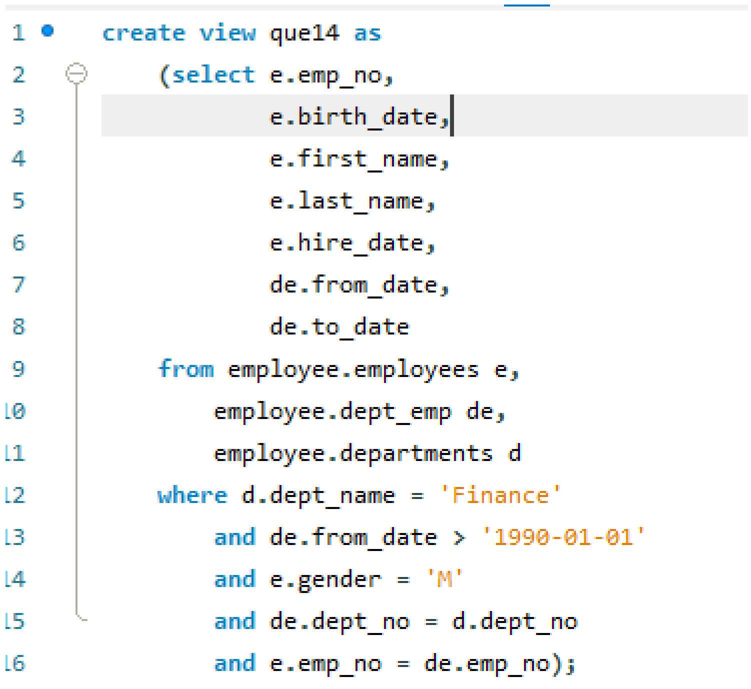


```sql
update employee.departments d
set d.dept_name = 'DB111'
where d.dept_no = 'd011'
```

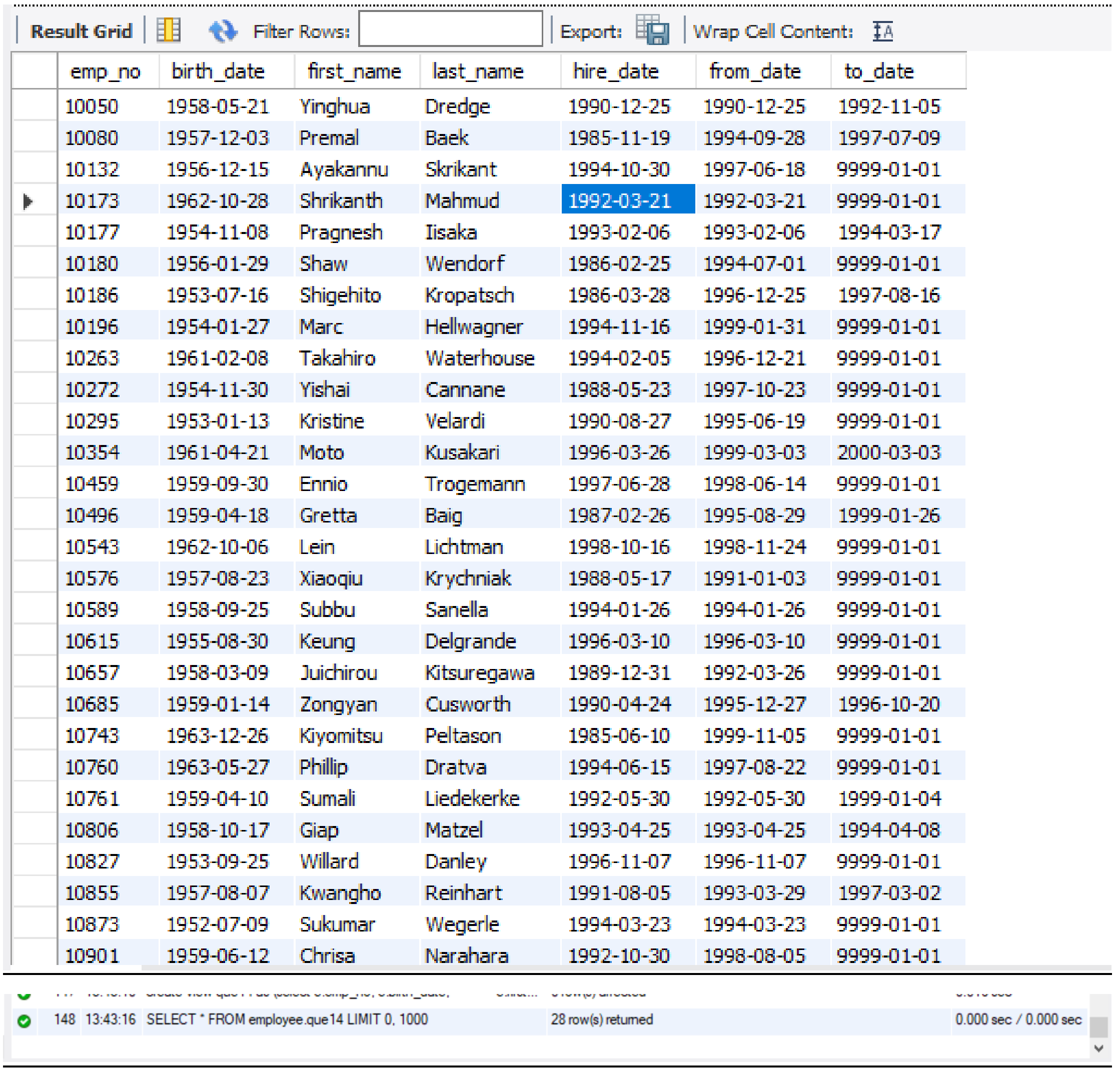


dept_no	dept_name
d001	Marketing
d002	Finance
d003	Human Resources
d004	Production
d005	Development
d006	Quality Management
d007	Sales
d008	Research
d009	Customer Service
d011	DB111
NULL	NULL

14. 新建视图，查询所有在 1990 年后入职过“Finance”部门的男员工信息，包括：emp_no, birth_date, first_name, last_name, hire_date, from_date, to_date。



```sql
create view que14 as
    (select e.emp_no,
            e.birth_date,
            e.first_name,
            e.last_name,
            e.hire_date,
            de.from_date,
            de.to_date
    from employee.employees e,
        employee.dept_emp de,
        employee.departments d
    where d.dept_name = 'Finance'
        and de.from_date > '1990-01-01'
        and e.gender = 'M'
        and de.dept_no = d.dept_no
        and e.emp_no = de.emp_no);
```



emp_no	birth_date	first_name	last_name	hire_date	from_date	to_date
10050	1958-05-21	Yinghua	Dredge	1990-12-25	1990-12-25	1992-11-05
10080	1957-12-03	Premal	Baek	1985-11-19	1994-09-28	1997-07-09
10132	1956-12-15	Ayakannu	Skrikant	1994-10-30	1997-06-18	9999-01-01
10173	1962-10-28	Shrikanth	Mahmud	1992-03-21	1992-03-21	9999-01-01
10177	1954-11-08	Pragnesh	Iisaka	1993-02-06	1993-02-06	1994-03-17
10180	1956-01-29	Shaw	Wendorf	1986-02-25	1994-07-01	9999-01-01
10186	1953-07-16	Shigehito	Kropatsch	1986-03-28	1996-12-25	1997-08-16
10196	1954-01-27	Marc	Hellwagner	1994-11-16	1999-01-31	9999-01-01
10263	1961-02-08	Takahiro	Waterhouse	1994-02-05	1996-12-21	9999-01-01
10272	1954-11-30	Yishai	Cannane	1988-05-23	1997-10-23	9999-01-01
10295	1953-01-13	Kristine	Velardi	1990-08-27	1995-06-19	9999-01-01
10354	1961-04-21	Moto	Kusakari	1996-03-26	1999-03-03	2000-03-03
10459	1959-09-30	Ennio	Trogemann	1997-06-28	1998-06-14	9999-01-01
10496	1959-04-18	Gretta	Baig	1987-02-26	1995-08-29	1999-01-26
10543	1962-10-06	Lein	Lichtman	1998-10-16	1998-11-24	9999-01-01
10576	1957-08-23	Xiaoqiu	Krychniak	1988-05-17	1991-01-03	9999-01-01
10589	1958-09-25	Subbu	Sanella	1994-01-26	1994-01-26	9999-01-01
10615	1955-08-30	Keung	Delgrande	1996-03-10	1996-03-10	9999-01-01
10657	1958-03-09	Juichirou	Kitsuregawa	1989-12-31	1992-03-26	9999-01-01
10685	1959-01-14	Zongyan	Cusworth	1990-04-24	1995-12-27	1996-10-20
10743	1963-12-26	Kiyomitsu	Peltason	1985-06-10	1999-11-05	9999-01-01
10760	1963-05-27	Phillip	Dratva	1994-06-15	1997-08-22	9999-01-01
10761	1959-04-10	Sumali	Liedekerke	1992-05-30	1992-05-30	1999-01-04
10806	1958-10-17	Giap	Matzel	1993-04-25	1993-04-25	1994-04-08
10827	1953-09-25	Willard	Danley	1996-11-07	1996-11-07	9999-01-01
10855	1957-08-07	Kwangho	Reinhart	1991-08-05	1993-03-29	1997-03-02
10873	1952-07-09	Sukumar	Wegerle	1994-03-23	1994-03-23	9999-01-01
10901	1959-06-12	Chrisa	Narahara	1992-10-30	1998-08-05	9999-01-01

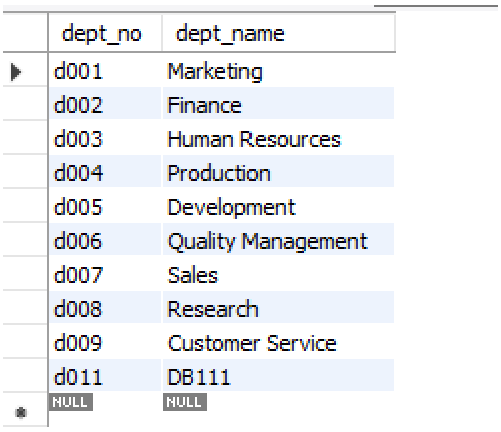
```
148  13:43:16  SELECT * FROM employee.que14 LIMIT 0, 1000    28 row(s) returned    0.000 sec / 0.000 sec
```

二、思考题

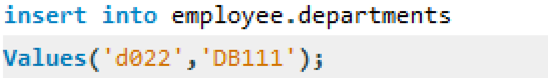
（PPT 中的思考题）

1. 如果 insert 一条数据到 Departments，但 dept_no 或 dept_name 和已有数据重复，会发生报错，如下：

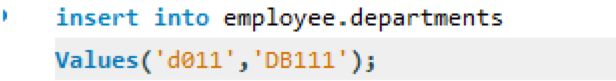
   Departments 表为：



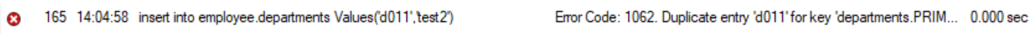
dept_no	dept_name
d001	Marketing
d002	Finance
d003	Human Resources
d004	Production
d005	Development
d006	Quality Management
d007	Sales
d008	Research
d009	Customer Service
d011	DB111
NULL	NULL



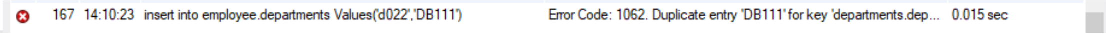
```sql
insert into employee.departments
Values('d022','DB111');
```



```sql
insert into employee.departments
Values('d011','DB111');
```



```
165  14:04:58  insert into employee.departments Values('d011','test2')        Error Code: 1062. Duplicate entry 'd011' for key 'departments.PRIM...   0.000 sec
```



```
167  14:10:23  insert into employee.departments Values('d022','DB111')       Error Code: 1062. Duplicate entry 'DB111' for key 'departments.dep...   0.015 sec
```

如果对 dept_emp 表新增数据，数据需满足以下条件：

1） emp_no 与已有数据不重复；

2） dept_no 的长度满足 char(4), emp_no 的长度满足 int；

3） 同一个 emp_no 的 from_date 和 to_date 形成的时间间隔之间不应该有重叠；

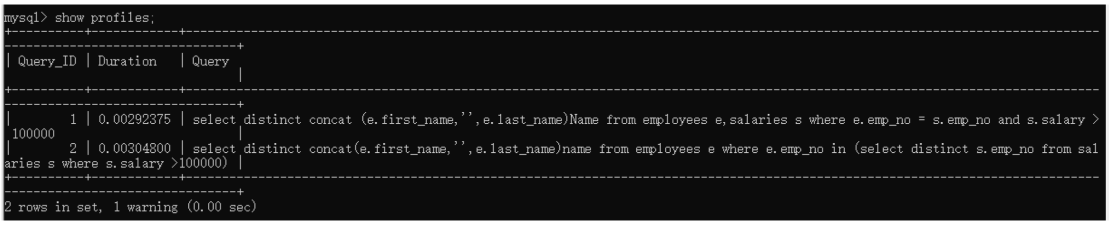
4） Emp_no 和 dept_no 必须分别存在于 employees 和 departments 表中；

5） 各项数据都不能为空。

保证数据正确性的机制：与其他表联系，与外键有关，保证 dept_emp 表中的 emp_no 和 dept_no 分别是 employees 表和 departments 表中的主键，并确保能有对应的数据。

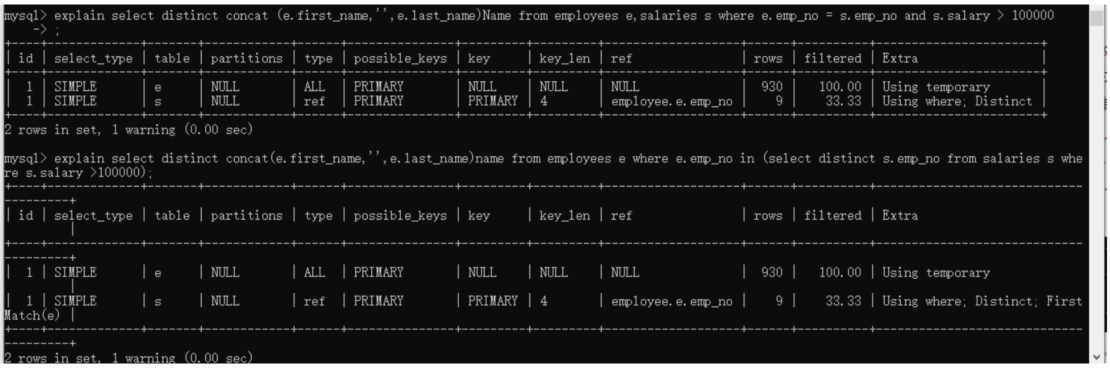
2. 使用 query profiler、Explain 进行 SQL 语句性能分析

使用 query profiler 分析：



```
mysql> show profiles;
+----------+------------+-----------------------------------------------------------------------------------------------------------------------------
--------------------------------+
| Query_ID | Duration   | Query
                                |
+----------+------------+-----------------------------------------------------------------------------------------------------------------------------
--------------------------------+
|        1 | 0.00292375 | select distinct concat (e.first_name,'',e.last_name)Name from employees e,salaries s where e.emp_no = s.emp_no and s.salary >
 100000                         |
|        2 | 0.00304800 | select distinct concat(e.first_name,'',e.last_name)name from employees e where e.emp_no in (select distinct s.emp_no from sal
aries s where s.salary >100000) |
+----------+------------+-----------------------------------------------------------------------------------------------------------------------------
--------------------------------+
2 rows in set, 1 warning (0.00 sec)
```

使用 explain 分析：

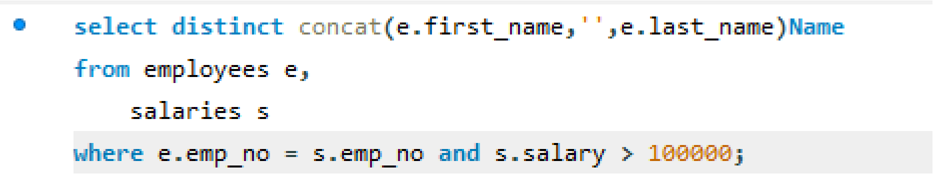


```
mysql> explain select distinct concat (e.first_name,'',e.last_name)Name from employees e,salaries s where e.emp_no = s.emp_no and s.salary > 100000
    -> ;
+----+-------------+-------+------------+------+---------------+---------+---------+--------------------+------+----------+------------------------+
| id | select_type | table | partitions | type | possible_keys | key     | key_len | ref                | rows | filtered | Extra                  |
+----+-------------+-------+------------+------+---------------+---------+---------+--------------------+------+----------+------------------------+
|  1 | SIMPLE      | e     | NULL       | ALL  | PRIMARY       | NULL    | NULL    | NULL               |  930 |   100.00 | Using temporary        |
|  1 | SIMPLE      | s     | NULL       | ref  | PRIMARY       | PRIMARY | 4       | employee.e.emp_no  |    9 |    33.33 | Using where; Distinct  |
+----+-------------+-------+------------+------+---------------+---------+---------+--------------------+------+----------+------------------------+
2 rows in set, 1 warning (0.00 sec)

mysql> explain select distinct concat(e.first_name,'',e.last_name)name from employees e where e.emp_no in (select distinct s.emp_no from salaries s whe
re s.salary >100000);
+----+-------------+-------+------------+------+---------------+---------+---------+--------------------+------+----------+----------------------------
---------+
| id | select_type | table | partitions | type | possible_keys | key     | key_len | ref                | rows | filtered | Extra
         |
+----+-------------+-------+------------+------+---------------+---------+---------+--------------------+------+----------+----------------------------
---------+
|  1 | SIMPLE      | e     | NULL       | ALL  | PRIMARY       | NULL    | NULL    | NULL               |  930 |   100.00 | Using temporary
         |
|  1 | SIMPLE      | s     | NULL       | ref  | PRIMARY       | PRIMARY | 4       | employee.e.emp_no  |    9 |    33.33 | Using where; Distinct; First
Match(e) |
+----+-------------+-------+------------+------+---------------+---------+---------+--------------------+------+----------+----------------------------
---------+
2 rows in set, 1 warning (0.00 sec)
```

可以看出第一条语句更快些。

第一条语句为：



```
select distinct concat(e.first_name,'',e.last_name)Name
from employees e,
    salaries s
where e.emp_no = s.emp_no and s.salary > 100000;
```